AIAA 99-0785

An O(Nm^2) Plane Solver for the Compressible Navier-Stokes Equations

J. L. Thomas
D. L. Bonhaus
W. K. Anderson
C. L. Rumsey
R. T. Biedron
NASA Langley Research Center
Hampton, Virginia 23681

37th Aerospace Sciences Meeting & Exhibit
January 11-14, 1999/Reno, NV

An O(Nm²) Plane Solver for the Compressible Navier-Stokes Equations

J. L. Thomas*
D. L. Bonhaus†
W. K. Anderson‡
C. L. Rumsey§
R. T. Biedron¶
NASA Langley Research Center
Hampton, Virginia 23681

1/11/99

Abstract

A hierarchical multigrid algorithm for efficient steady solutions to the two-dimensional compressible Navier-Stokes equations is developed and demonstrated. The algorithm applies multigrid in two ways: a Full Approximation Scheme (FAS) for a nonlinear residual equation and a Correction Scheme (CS) for a linearized defect correction implicit equation. Multigrid analyses which include the effect of boundary conditions in one direction are used to estimate the convergence rate of the algorithm for a model convection equation. Three alternating-line-implicit algorithms are compared in terms of efficiency. The analyses indicate that full multigrid efficiency is not attained in the general case; the number of cycles to attain convergence is dependent on the mesh density for high-frequency cross-stream variations. However, the dependence is reasonably small and fast convergence is eventually attained for any given frequency with either the FAS or the CS scheme alone. The paper summarizes numerical computations for which convergence has been attained to within truncation error in a few multigrid cycles for both inviscid and viscous flow simulations on highly stretched meshes.

Introduction

There has been an explosive growth in the use of computational fluid dynamics methods in the aircraft design cycle over the past twenty-five years. Recently there has been an emphasis on three-dimensional Navier-Stokes simulations over complex configurations; computations with 10–20 million grid points are commonplace in focused applications. Even with the advent of more powerful computers, algorithms that attain optimal convergence rates are important to enabling these computations to be accomplished in a reasonable wall-clock time. An optimal method is one in which the arithmetic operations to attain a solution to within truncation error scale as $O(N)$, where N is the number of equations to be solved. Here, N is the number of finite volumes in the solution (N_{FV}) times the number of conservation equations for each finite volume (m); i.e., $N = m\,N_{FV}$. Generally, the total operation count can be expressed as $c\,N\,W_{MWU}$ where W_{MWU} is the operation count corresponding to one minimal work unit (MWU), i.e., the simplest possible discretization of the equations to the order desired, and c is a constant that differentiates one optimally converging method from another.[1] One method of attaining optimal convergence rates is the multigrid method. For elliptic equations, textbook efficiencies, which attain convergence in four to five residual evaluations, are possible.[1,2] For hyperbolic equations, $O(N)$ methods have been developed for the incompressible Euler equations[1,3,4] and for compressible Euler equations using either the Full Approximation Scheme (FAS)[5,6] or the defect correction (DC) scheme.[7–10] Multigrid solvers for viscous flows have also been developed using these approaches.[11–14]

For the compressible Navier-Stokes equations, textbook efficiencies have not been attained for general situations; the barriers which need to be over-

*Head, Aerodynamic and Acoustic Methods Branch (AAMB), AIAA Fellow.
†Member AAMB, AIAA Senior Member.
‡Member AAMB, AIAA Associate Fellow.
§Member AAMB, AIAA Senior Member.
¶Member AAMB

come are addressed by Brandt.[2] One of the principal difficulties for complex-geometry applications has been the need for very highly stretched grids to resolve viscous flows near the body, with correspondingly bad (and unintended) aspect ratios in other regions. As an example, for a transonic wing with separated flow, the convergence rate for a widely used multigrid method based on the FAS algorithm and a multistage Runge-Kutta scheme[13] requires on the order of 1500 residual evaluations to attain convergence of the lift to within one percent of the asymptotic value.[15] Recently, improvements that use line-implicit algorithms and semi-coarsening approaches have been demonstrated for these applications.[16–19]

The purpose of this paper is to introduce an algorithm that uses full-coarsening multigrid to accelerate convergence for viscous applications. All of the basic elements of the method (multigrid, line relaxation, upwind differencing, defect correction, etc.) are well known. The algorithm applies the FAS scheme to the nonlinear residual equations and the correction scheme (CS) multigrid to the linearized implicit equations. The methodology uses alternating-line-implicit methods, although the essential feature is the requirement for the solution to a local block Jacobian matrix of size m at each grid point. Thus the operation counts are $O(N_{FV}m^3)$ and the computational work will scale as $O(Nm^2)$. This is in contrast to the class of $O(N)$ algorithms discussed by Brandt[2] which decouple the equations into separate scalar contributions, each of which is treated optimally. The computations are supported by analysis of convergence for a simple model convection problem which shows many of the essential features of the resulting algorithm. Comparisons are made with the baseline solver implemented in a widely used production code CFL3D.[14] Because the algorithm contains many elements of existing methods, the methodology should be able to be incorporated into production codes and to accelerate convergence for realistic applications.

Baseline Method

The steady-state results are obtained with a finite-volume approach based on an upwind-biased treatment of the convective and pressure (Euler) terms and central differencing for the viscous terms. The method has been implemented in the CFL3D code, used widely for large-scale computations and described by Krist, et al.[14] Only a few basic features are cited here. The Riemann interface solver is the flux-difference-splitting method and the κ-scheme of Van Leer[20] is used for state-variable extrapolations. Convergence to steady state is accelerated through the Full MultiGrid (FMG) process, i.e., mesh sequencing and FAS multigrid with an approximately factored implicit method as the relaxation scheme. In the approximate factorization (AF) method, the full matrix is replaced with a sequence of simpler operators, each of which is a block tridiagonal or pentadiagonal operator. However, the baseline AF method is used almost exclusively in its diagonal form.[21] The multigrid method was demonstrated to yield grid-independent convergence rates for Euler simulations using the flux-vector splitting method by Anderson, et al.[5] The scheme has been used routinely to accelerate the solution for viscous flow using flux-vector splitting methods for the dissipation, with a spectral radius approximation to the viscous Jacobians. The scheme is generally applied as a W(1,0)[5] FAS cycle using a Courant number of 5. For time-dependent simulations, because of the severe time-step limitation of the method, subiterations have been used to improve the accuracy and stability of the implicit scheme, as in Rumsey, et al.[22]

Multigrid Method

The present algorithm uses multigrid in two ways. The first way is through an outer FAS multigrid cycle to solve the second-order-accurate, nonlinear steady-state residual operator. The second way is through an inner iteration to solve the first-order-accurate, linearized implicit operator. We describe the multigrid methods for the two approaches below by means of a two-grid approach, in which the fine grid is denoted by superscript h and the coarse grid by superscript $2h$. The coarser grid equations are themselves solved with γ cycles of the algorithm applied recursively, where $\gamma = 1$ corresponds to a V-cycle and $\gamma = 2$ to a W-cycle.

FAS Multigrid Cycle

The second-order-accurate steady-state residual operator to be solved on the finest grid is defined as

$$\mathbf{R}^h(\mathbf{Q}^h) = 0 \tag{1}$$

where this equation represents the inviscid convective and pressure terms and the viscous diffusion and heat transfer terms; $\mathbf{Q}^h$ represents a vector of size m at each of the N_{FV} finite-volumes in the domain. After relaxation(s) of the fine-grid operator to obtain $\tilde{\mathbf{Q}}^h$, the coarse-grid equation at level $2h$ to be solved for a correction to the fine grid is

$$\mathbf{R}^{2h}(\mathbf{Q}^{2h}) = \mathbf{R}^{2h}(\mathcal{R}\tilde{\mathbf{Q}}^h) - \mathcal{R}\mathbf{R}^h(\tilde{\mathbf{Q}}^h) \tag{2}$$

where $\mathcal{R}$ denotes a restriction operator for transfer of information to the coarser grid and the $\tilde{}$ superscript denotes a most recently available value. The correction from the coarser grids is prolonged to the finer grids as

$$\tilde{\mathbf{Q}}^h \leftarrow \tilde{\mathbf{Q}}^h + \mathcal{P}(\mathbf{Q}^{2h} - \mathcal{R}\tilde{\mathbf{Q}}^h) \tag{3}$$

where $\mathcal{P}$ denotes a prolongation operator. The FAS cycle described above is used extensively for current Euler and Navier-Stokes solvers. The differences in convergence between solvers lie chiefly in their choice of relaxation (smoothing) scheme, such as the approximate-factorization method[5] or multistage Runge-Kutta methods with implicit residual smoothing.[6,13]

From the standpoint of a Newton method, the fine-grid correction can be written

$$[\frac{\partial \mathbf{R}^h}{\partial \mathbf{Q}^h}]\,(\Delta\tilde{\mathbf{Q}}^h) = -\,\mathbf{R}^h(\tilde{\mathbf{Q}}^h) \tag{4}$$

where the solution is updated as $\tilde{\mathbf{Q}}^h \leftarrow \tilde{\mathbf{Q}}^h + \Delta\tilde{\mathbf{Q}}^h$. The implicit equation is a large-banded matrix equation which is usually approximated for solution with two approaches. The first approach is to use an approximate linearization of the residual; commonly, the linearization of the residual uses first-order discretizations for the convective and pressure contributions. For this approach, we can write the implicit scheme as

$$[\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}]\,(\Delta\tilde{\mathbf{Q}}^h) = -\,\mathbf{R}^h_t(\tilde{\mathbf{Q}}^h) \tag{5}$$

where the subscripts t and d denote some desired "target" and "driver" schemes on the right and left sides, respectively, of the equation. Note that this equation is the defect correction form of the equations[7,8,10] written in delta form for the update. This "defect" in the implicit approximation leads to some interesting consequences for the algorithm, as discussed subsequently, even if we make no further approximations.

The second approach is to solve the full matrix equation iteratively or with a noniterative approximate factorization. For instance, Anderson, et al.[11] used red-black block-matrix subiterations (usually 15) with point (m-block matrix inversions) relaxations to approximate the solution of the fine-grid implicit equation for unstructured grids. Thus the efficacy of the solver becomes a trade-off between the additional work of the subiterations and the reduced approximations to the implicit equation.

If the implicit terms are differenced with first-order-accurate upwind discretizations, the resulting equations are block diagonally dominant. Therefore, these equations can be solved efficiently with multigrid methods and standard relaxation methods used for solution of iterative equations, such as Jacobi and Gauss-Seidel relaxation. With a second-order-accurate discretization, the implicit equations are only block diagonally dominant for a CFL number of unity. In either case, because the implicit equations are linear, a CS multigrid method can be used, as described below.

CS Multigrid Cycle

During the iterative process to solve the linear implicit equation above, the second-order accurate residual is held fixed, defined as $\mathbf{b}^h = -\mathbf{R}^h_t(\tilde{\mathbf{Q}}^h)$. The equations are first relaxed on the fine grid for an approximate solution using a subiteration counter l, $(l = 0, 1, \ldots, N_s - 1)$, as

$$[\overline{\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}}]\,[(\Delta\tilde{\mathbf{Q}}^h)^{l+1} - (\Delta\tilde{\mathbf{Q}}^h)^l] = \mathbf{b}^h - [\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}](\Delta\tilde{\mathbf{Q}}^h)^l \tag{6}$$

or equivalently,

$$[\overline{\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}}]\,(\Delta\tilde{\mathbf{Q}}^h)^{l+1} = \mathbf{b}^h - [\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h} - \overline{\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}}](\Delta\tilde{\mathbf{Q}}^h)^l \tag{7}$$

where the approximation to the implicit equation on the left side is denoted with an overline notation and $(\Delta\tilde{\mathbf{Q}}^h)^0 = 0$. The coarse grid also supplies a correction to the linear system through solution of the coarse grid correction equation, defined below.

$$\frac{\partial \mathbf{R}^{2h}{}_d}{\partial \mathbf{Q}^{2h}}[\Delta\Delta\mathbf{Q}^{2h}] = \mathcal{R}[\mathbf{b}^h - [\frac{\partial \mathbf{R}^h{}_d}{\partial \mathbf{Q}^h}](\Delta\tilde{\mathbf{Q}}^h)] \tag{8}$$

where the latest value of $(\Delta\tilde{\mathbf{Q}}^h)$ is used on the right. The correction to $\Delta\mathbf{Q}^{2h}$ from the coarser grid is prolonged to the finer grids as

$$\Delta\tilde{\mathbf{Q}}^h \leftarrow \Delta\tilde{\mathbf{Q}}^h + \mathcal{P}[\Delta\Delta\mathbf{Q}^{2h}] \tag{9}$$

All of the boundary conditions are completely linearized and incorporated into the defect correction

operator. In the CS multigrid, the coarser grid implicit matrices are found by restricting the corresponding finer grid implicit matrix contributions with the result that the above linearization need only be done for the fine grid. The linearization includes a time term which can be ramped from small values at impulsive starts from freestream conditions to Courant numbers on the order of 300–500. The linear system is easier to solve if the time step is small, and there is little to be gained in the second-order residual convergence for Courant numbers beyond these values for most flows.

Convergence of Defect Correction

The DC method can be written in terms of target and driver operators L_t and L_d, respectively, on a given grid h as

$$L_d(u^{n+1}) = L_d(u^n) - L_t(u^n) \tag{10}$$

where the operators are designed to approximate the actual partial differential operator to within an order property

$$L_d = L + \tau_d = L + O(h) \tag{11}$$

$$L_t = L + \tau_t = L + O(h^2) \tag{12}$$

Now substituting from the above into Eq. (10), defining $u = u_{\text{exact}} + e$ where $L(u_{\text{exact}}) = 0$ and e is the truncation error, then

$$\begin{aligned}(L+\tau_d)(e^{n+1}) &= (L+\tau_d)(e^n) - L(e^n) + O(h^2) \\ &= \tau_d(e^n) + O(h^2)\end{aligned} \tag{13}$$

If we assume that $(L+\tau_d)^{-1} = L^{-1} + O(h)$, then

$$e^{n+1} = L^{-1}\tau_d(e^n) + O(h^2) \tag{14}$$

Telescoping the error terms from an arbitrary starting error e^0, then

$$L(e^n) = L(L^{-1}\tau_d)^n(e^0) + O(h^2) \tag{15}$$

Because we need only converge the solution until truncation error, $L(e^n) = O(h^2)$, we can categorize the convergence of defect correction into three regimes. The first regime corresponds to a mesh fine enough that the first-order scheme satisfies the order property above, for which convergence would be expected in just a few iterations. The second regime corresponds to a mesh for which the second-order scheme satisfies the order property but the first-order scheme does not. It is this regime for which the slowest convergence would be expected. The third regime corresponds to a mesh coarse enough that neither the first-order nor second-order scheme satisfies the order property above; in this regime, convergence to truncation error is nonetheless generally rapid because the truncation errors are $O(1)$.

These regimes can be classified sharply for the κ family if we consider convection, $Lu \equiv u_x + tu_y$,

$$\tau_d(u) = -\frac{h}{2}[u_{xx} + tu_{yy}] + O(h^2) \tag{16}$$

$$\tau_t(u) = \frac{h^2(\kappa - 1/3)}{4}[u_{xxx} + tu_{yyy}] + O(h^3) \tag{17}$$

Considering an exact solution with cross-stream frequency ω, corresponding to $u = \exp(i\omega(y - tx))$, the scheme attains its design accuracy when

$$|\tau_d| = |\frac{ht}{2}\omega^2(1+t)| \le \frac{1}{2} \tag{18}$$

$$|\tau_t| = |\frac{h^2 t(\kappa - 1/3)}{4}\omega^3(t^2-1)| \le \frac{1}{4} \tag{19}$$

The inequalities on the right denote approximate values observed in parametric calculations; for values satisfying the inequality, the scheme attains its design accuracy—an asymptotic error reduction of 2^p as the mesh is refined by a factor of two. Thus the three regions can be classified as below:

$$\begin{aligned}\text{I}: &\quad |\tau_d| \le \tfrac{1}{2} \\ \text{II}: &\quad |\tau_d| \ge \tfrac{1}{2} \text{ and } |\tau_t| \le \tfrac{1}{4} \\ \text{III}: &\quad |\tau_t| \ge \tfrac{1}{4}\end{aligned}$$

The three regions denote the disparity between the dissipation of the driver and target schemes and are a restatement of the "survival distances" associated with convection schemes derived by Brandt and Yavneh[4] in studies of the incompressible Navier-Stokes equations, for standard schemes of the type considered here as well as for hybrid schemes of improved accuracy and convergence.

Analysis of Convergence

Model Problem

To analyze convergence, we consider a model convection equation corresponding to flow at some angle of attack α to a unit square,

$$s_x + t\, s_y = 0, \qquad x \in [0,1], y \in [0,1] \tag{20}$$

where $t = \tan\alpha$. We consider a finite-volume scheme for Eq. (20) on a uniform Cartesian grid with spacings $h_x = 1/N_x$ and $h_y = 1/N_y$ in the x- and y-directions, respectively, as

$$\delta_x^h s^h + t\, \delta_y^h s^h = 0 \tag{21}$$

where $s^h_{j,k}$ is defined at the locations $x_j = (j-1/2)h$ and $y_k = (k-1/2)h$ for $(j,k) = 1,2,\ldots,(N_x,N_y)$. Assuming periodicity of the solution in y, $s^h_{j,k} = u^h_j \exp(i\,\omega_y\, y_k)$ where ω_y is a given frequency in the y-direction. The exact solution to Eq. (20) is $s(x,y) = f(\xi)$ where $\xi = y - t\,x$, corresponding in this case to $[u_j]_{\text{exact}} = \exp(-i\,t\,\omega_y x_j)$. The set of discrete frequencies ω_y realizable on a given grid can be defined in terms of $\theta_y = \omega_y\, h_y$ as

$$\theta_y = \frac{2\pi l}{N_y}, \qquad l = 0,1,\ldots,N_y - 1 \tag{22}$$

The numerical scheme can be written in matrix notation as

$$(\delta_x^h + t\,\hat{\delta}_y^h) u^h \equiv L^h\, u^h = -f^h \tag{23}$$

where u^h is the vector of unknowns at the interior points and f^h is the vector associated with boundary conditions imposed at the inflow from the exact solution. We consider either schemes of first-order accuracy or one of the family of κ schemes, $\kappa \in [-1,1]$. As a shorthand notation, we refer to the first-order scheme as $\kappa = -3$. The matrices are

$$\delta_x^h = \frac{1}{h_x}\begin{bmatrix} c_3 & c_4 & 0 & 0 & 0 & 0 \\ c_2 & c_3 & c_4 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_4 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & \tilde{c}_1 & \tilde{c}_2 & \tilde{c}_3 \end{bmatrix} \tag{24}$$

$$f^h = \frac{1}{h_x}\begin{bmatrix} c_2 \exp(-i\,t\,\omega_y\, x_0) + c_1 \exp(-i\,t\,\omega_y\, x_{-1}) \\ c_1 \exp(-i\,t\,\omega_y\, x_0) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{25}$$

$$\hat{\delta}_y^h = \frac{1}{h_y}[c_1 \exp(-i2\theta_y) + c_2 \exp(-i\theta_y) + c_3 + c_4 \exp(i\theta_y)]I \tag{26}$$

where $\hat{\delta}_y^h$ is a diagonal matrix and δ_x^h is lower triangular for the fully upwind schemes, $\kappa \subset \{-3,-1\}$. The coefficients for $\kappa \in [-1,1]$ are

$\{c_1,c_2,c_3,c_4\} = \frac{1}{4}\{1-\kappa,\ 3\kappa-5,\ 3(1-\kappa),\ 1+\kappa\}$

and for $\kappa = -3$ are

$\{c_1,c_2,c_3,c_4\} = \{0,-1,\ 1,\ 0\}$

and $\tilde{c}_{1-3}$ reflect the incorporation of nonreflecting boundary conditions at the right for the upwind-biased schemes, $\kappa \in (-1,1]$.

Defect Correction

The defect correction scheme to solve this equation can be written in delta form, $\Delta u = u^{n+1} - u^n$, where the superscript n denotes a cycle (iteration) counter, as

$$L_d^h(\Delta u^h) = -L_t^h (u^h)^n - f_t^h \qquad n = 0,1,\ldots,N_c - 1 \tag{27}$$

Defining the algebraic error e^h as the difference between the exact discrete solution and the current approximation, $u^h = u^h_{\text{exact}} + e^h$, then Eq. (27) can be written

$$L_d^h(\Delta e^h) = -L_t^h (e^h)^n \quad \text{for n} = 0,1,\ldots,\mathrm{N_c} - 1 \tag{28}$$

Subiteration Scheme

An exact solution of this equation at each cycle would correspond to an unfactored solver and is easily accomplished here with any of the fully-upwind target schemes. However, as a model for the process when we solve the coupled system of Euler/Navier-Stokes equations, we consider an approximate solution of the driver operator, including the effect of subiterations.

$$\overline{L_d^h}[(\Delta e^h)^{l+1} - (\Delta e^h)^l] = -L_t^h (e^h)^n - L_d^h (\Delta e^h)^l \quad \text{for l} = 0,1,\ldots,\mathrm{N_s} - 1 \tag{29}$$

where $N_s = 1$ corresponds to a noniterative approximation, Eq. (28) is solved if the subiteration process converges, and $(\Delta e^h)^0$ is taken as zero at the start of the process. In addition to the unfactored scheme, we consider three approximations to L_d^h, two of which are an Alternating Line-Jacobi (ALJ) and a spatially-factored Approximate Factorization (AF) scheme (also see Appendices I-II)

$$\begin{aligned} \overline{L^h}|_{ALJ} &= (\frac{I}{\Delta t} + \delta_x^h + t\,(\hat{\delta}_y^h)^D) \\ &\quad (\frac{I}{\Delta t} + t\,(\hat{\delta}_y^h)^D + (\delta_x^h)^D)^{-1} \\ &\quad (\frac{I}{\Delta t} + t\,\hat{\delta}_y^h + (\delta_x^h)^D) \end{aligned} \tag{30}$$

$$\overline{L^h}|_{AF} = (\frac{I}{\Delta t} + \delta_x^h)(\frac{I}{\Delta t})^{-1}(\frac{I}{\Delta t} + t\,\hat{\delta}_y^h) \tag{31}$$

where superscript D denotes the diagonal contribution from the matrix arising from discretization in x and y; for the differencing in y, the diagonal term is the $\exp(0)$ contribution to the difference symbol, Eq. (26). The third approximation to L_d^h is the Alternating Line Red-Black (ALRB) scheme (see Appendix III), which can be represented as

$$\overline{\mathbf{L}^h}|_{ALRB} = (\mathbf{N}_y^{-1} + \mathbf{N}_x^{-1} - \mathbf{N}_y^{-1}\mathbf{L}_d^h\mathbf{N}_x^{-1})^{-1} \tag{32}$$

where $\mathbf{N}_y$ and $\mathbf{N}_x$ represent sweeps through the mesh in which the y and x lines are solved implicitly in a red-black fashion, respectively. The equations use an artificial time term defined in terms of the CFL number as

$$\frac{I}{\Delta t} = (\frac{I}{h_x} + t\,\frac{I}{h_y})/CFL \tag{33}$$

where the time term serves principally as a relaxation device for steady-state solutions. For all of the analyses here, we use $CFL = \infty$ for the ALJ and the ALRB schemes.

Error Amplification Matrices

It is usual to construct the amplification of the error for the red-black scheme or the multigrid scheme considering two frequencies at a time, ω_y and $\omega_y + \pi/h_y$. Thus the original error distribution

$$\begin{bmatrix} (e_1^h)^0\,\exp(i\omega_y y) \\ (e_2^h)^0\,\exp(i(\omega_y + \frac{\pi}{h_y})y) \end{bmatrix} \tag{34}$$

becomes

$$\begin{bmatrix} (e_1^h)^n\,\exp(i\omega_y y) \\ (e_2^h)^n\,\exp(i(\omega_y + \frac{\pi}{h_y})y) \end{bmatrix} \tag{35}$$

where

$$\begin{bmatrix} (e_1^h)^n \\ (e_2^h)^n \end{bmatrix} = \mathbf{G}^{[n]} \begin{bmatrix} (e_1^h)^0 \\ (e_2^h)^0 \end{bmatrix} \tag{36}$$

For single-grid relaxations, the matrix $\mathbf{G}$ is a diagonal matrix; off-diagonal entries arise for red-black relaxations and for intergrid transfers of information. The error amplification matrix per cycle for a multigrid cycle with ν_1 relaxations before restriction, ν_2 relaxations after prolongation, and an exact solution on the coarser mesh $2h$ can be written as

$$\mathbf{G} = \mathbf{S}^{[\nu_2]}\,[\mathbf{I} - \mathcal{P}\,(\mathbf{L}_t^{2h})^{-1}\,\mathcal{R}\,\mathbf{L}_t^h]\,\mathbf{S}^{[\nu_1]} \tag{37}$$

where

$$\mathbf{L}_t^h = \begin{bmatrix} L_t^h(\theta_y) & 0 \\ 0 & L_t^h(\theta_y + \pi) \end{bmatrix} \tag{38}$$

$$\mathcal{R} = \begin{bmatrix} R\cos(\theta_y/2) & R\sin(\theta_y/2) \end{bmatrix} \tag{39}$$

$$\mathcal{P} = \begin{bmatrix} P(3\cos(\theta_y/2) + \cos(3\theta_y/2))/4 \\ P(3\sin(\theta_y/2) - \sin(3\theta_y/2))/4 \end{bmatrix} \tag{40}$$

The restriction and prolongation matrices in x are defined in Appendix IV. An alternate formulation of the amplification matrix is defined as

$$\mathbf{G}^* = \mathcal{R}^*\,\mathbf{G}\mathcal{P}^* \tag{41}$$

where R^* and P^* are higher order restriction and prolongation operators, also defined in Appendix IV, and the matrices $\mathcal{R}^*$ and $\mathcal{P}^*$ are constructed assuming no aliasing errors occur in the transfer of information in the periodic direction y.

$$\mathcal{R}^* = \begin{bmatrix} R^* & 0 \end{bmatrix};\ \mathcal{P}^* = \begin{bmatrix} P^* \\ 0 \end{bmatrix}$$

In this formulation, even for the single-grid scheme, the amplification matrix accounts for the transfer of information between the coarser and finer mesh. The equation represents the prolongation of fine grid low-frequency errors on the coarser mesh to the fine mesh, their amplification on the fine mesh, and a

subsequent restriction down to the coarser mesh. The alternate formulation reduces the size of the matrix eigenvalue problem to be solved from $2N_x \times 2N_x$ to $N_x/2 \times N_x/2$ and generally yields results which are within 1 cycle of the standard formulation. Larger differences arose when the maximum amplifications occured near the $\theta_y/\pi = 0.5$ boundary; these differences have not been resolved, but in those instances where differences do occur, the standard amplification matrix is used.

The relaxation matrix $\mathbf{S}$ accounts for $l \equiv N_s$ subiterations as

$$\mathbf{S} = \Sigma^l \Phi + \Xi^{[l]} \tag{42}$$

$$\Sigma^l = \Xi^{[l-1]} + \Xi^{[l-2]} + \ldots + \Xi^{[1]} + \mathbf{I} \tag{43}$$

$$\Phi = (\overline{\mathbf{L}_d^h})^{-1} (\mathbf{L}_d^h - \mathbf{L}_t^h) \tag{44}$$

$$\Xi = \mathbf{I} - \overline{\mathbf{L}_d^h}^{-1} \mathbf{L}_d^h \tag{45}$$

where the superscript $[n]$ on a term denotes the term raised to the nth power when there might be confusion with the n or l superscript. The matrix corresponding to amplification of the discrete residual can be written as

$$\mathbf{H} = \mathbf{L}_t^h \, \mathbf{G} \, (\mathbf{L}_t^h)^{-1} \tag{46}$$

Error Bounds

The usual bounds considered for amplification of errors are the spectral norm and the L_2 norm. The spectral norms of $\mathbf{G}$ and $\mathbf{H}$ are the same because they are related by a similarity transformation, although we find in practice that the computation of $\rho(\mathbf{H})$ is subject to round-off errors. For elliptic equations, Brandt[23] has shown that these bounds are attained for general domains using local mode (fully periodic) analyses as long as the cycle is supplemented with additional (and negligibly small) processing at and near the boundary. For hyperbolic equations, the boundary has a more global influence; in the case of convection, information propagates from the boundary into the domain. For convective equations, we find the spectral norm to be not very useful, since it often is reached only after a large number of iterations and may not be observed in practice (see Appendix V). Thus we concentrate on the L_2 norm of either the error or the residual and correlate the number of cycles to reduce them by specified amounts.

Stopping Criteria

By far, the most important bound is that for the error, because we need only converge the numerical scheme to within some measure of truncation error on each grid. If we reduce the algebraic error on each mesh by a factor f, the total error on subsequent meshes can be expressed as $u^h = u_{\text{exact}} + \tau^h (1 + \beta)$

$$\beta = \frac{(2^p - 1)}{f} (1 + \frac{2^p}{f} + (\frac{2^p}{f})^2 + \ldots) \tag{47}$$

If we somewhat arbitrarily invoke that $\beta = 1/8$, so that the algebraic error is 1/8th of the truncation error, then the error reduction values f are 10 and 28 for the $p = 1$ (first-order) and $p = 2$ (second-order) schemes, respectively. We apply this criteria for the local error to the L_2 error norms. In the development of Eq. (47), we assume that the prolongation operator used in the interpolation of the solution from the coarser mesh is nearly unity, consistent with a high-order interpolation. If we combine that with a high-order restriction operator, we can enforce that a given fine-grid algebraic error as interpolated to the coarse mesh be reduced by the factor f through examination of the $\mathbf{G}^*$ amplification matrix given above. In practice, the residual is usually monitored, and grid-independent convergence rates for the residual have been used as an assessment of whether the multigrid is functioning properly. Hence, we show some results for reduction of the residual norms.

Efficiency Measure

In all cases, we show the number of updates of the solution on the fine grid, defined as

$$N_U = N_c N_s (\nu_1 + \nu_2)$$

as the efficiency measure. This is only approximate, because it only accounts directly for the line inversions done through either subiterations or additional multigrid relaxations. The residual evaluations and any work done on coarser grid are not taken into account. We can also define an effective norm as the norm per update of the solution, i.e.,

$$(||\mathbf{G}^{[N_c]}||_2)_{effective} = (||\mathbf{G}^{[N_c]}||_2)^{1/N_U} \tag{48}$$

With the above models, there are a large number of possible parametric studies of the type that have been conducted extensively for elliptic equations; the

general approach is extendable to systems of equations, three dimensions, and semi-coarsening multigrid schemes. We explore only a few parameters below, focusing on the effect of convergence for the second-order-accurate discretizations, including the effects of relaxation scheme, subiterations, and FAS multigrid on the convergence.

Convergence Results

Unfactored Scheme

The number of updates for convergence with the unfactored scheme is considered an upper bound for performance with defect correction. The direct solution of the associated large-bandwidth equation is not viable from an efficiency standpoint, but with the first-order implicit equations, convergence can be attained in only a few multigrid cycles.[7–9] For the unfactored scheme, we show results with $\kappa_t = 0$ and $\kappa_d = -3$ for a typical case, $\alpha = 30\,\text{deg}$, both with and without multigrid. We examine the number of cycles to reach convergence of $||\mathbf{G}^*||_2$ below $1/28$, with θ_y over all possible discrete frequencies. The results are shown in Table 1. There is a growth in the number of cycles in either case because of the defect correction approximation, i.e., the disparity between the target second-order and the driver first-order scheme. The growth is, however, not explosive. The multigrid scheme is effective, even for this unfactored implicit scheme, in reducing both the error and the residual norms in comparison to the single-grid scheme. The improvement approaches a factor of two as the grid is refined. On the finest mesh, the number of cycles to convergence for the lowest frequency considered and its associated harmonic, $\theta_y/\pi = (-0.0156, 0.9844)$, was five for the single-grid scheme and four for the multigrid scheme; the slowest convergence occurred at frequencies of $\theta_y/\pi = (-0.25, 0.75)$ and $\theta_y/\pi = (-0.343, 0.657)$, respectively, for the two schemes. The average number of cycles over all the frequencies considered was 15.6 and 8.8 for the single-grid and multigrid schemes, respectively. Results obtained by monitoring convergence of $||\mathbf{H}||_2$ below 10^{-4} showed a similar growth in the number of cycles for both single and multigrid schemes, although the multigrid showed only a 20-percent improvement in the number of cycles to attain this level of residual convergence on the finest mesh.

The above comparison represents a worst case scenario for the bound, because all frequencies need to be reduced by a constant amount. In fact, through the FMG process, the troublesome frequencies may have very small amplitudes in the starting solutions for a given grid. Additionally, those frequencies which are unresolved need only be reduced by less than $1/28$ to be within truncation error. Thus, in Table 2, we show the number of updates considering only a single frequency, corresponding to $\omega_y = 8\pi$. We also show τ values from Eq. (18–19) in the table; the $N_x = N_y = 32$ grid is the first grid that would show close to the desired order property of a factor of four reduction in the error as the grid is refined by a factor of two in the x and y directions. The first order scheme does not provide resolution over the entire domain except on the very finest mesh; note the span of grids between accuracy of the first order and second order operators. The asymptotic convergence is quite good, as predicted by the defect correction asymptotic analysis, because eventually the solution is resolved even with the first-order driver scheme. In this limit, the multigrid is neither effective nor needed, as the single grid converges the error and residual in a few cycles. The boundary between regions II and III exhibits the slowest behavior; the multigrid method improves the convergence because the truncation error of the coarser mesh target operator is better than the truncation error of the fine mesh driver operator. Although not shown, results with the $\kappa_t = -1$ scheme were similar to those above, although the number of cycles were 30-50 percent lower.

N_x	Single Grid	Multigrid
8	7	7
16	9	8
32	12	10
64	18	12
128	25	13

Table 1. Number of updates, N_U, to reduce error norm $||\mathbf{G}^*||_2 \leq 1/28$ for the unfactored DC scheme; $|\theta_y| \leq 2\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

N_x	Single Grid	Multigrid	$\lvert\tau_d\rvert$	$\lvert\tau_t\rvert$
8	3	4	36	7
16	8	8	18	2
32	12	7	9	0.5
64	10	4	4	0.1
128	6	4	2	0.03
256	5	4	1	0.007

Table 2. Number of updates, N_U, to reduce error norm $||\mathbf{G}^*||_2 \leq 1/28$ for the unfactored DC scheme; $\omega_y = 8\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

Comparisons with the AF Scheme

For the noniterative ($N_s = 1$) AF scheme, we first show results with $\kappa_d = \kappa_t = -3$ for $\alpha = 45\,\text{deg}$,

Method	N_x	$\|\|\mathbf{G}\|\|_2$	$\|\|\mathbf{H}\|\|_2$	$\rho(\mathbf{G})$
SG	8	.91	.91	.5
SG	16	.97	.97	.5
SG	32	.99	.99	.5
SG	64	.99	.99	.5
MG	8	.67	.64	.57
MG	16	.82	.73	.57
MG	32	.98	.86	.58
MG	64	1.13	1.01	.58

Table 3. Norms after one update for the AF scheme; $|\theta_y| \leq 2\pi, \kappa_t = \kappa_d = -3, \alpha = 30\,\text{deg}$

N_x	SG Pred.	SG CFL3D	MG Pred.	MG CFL3D
8	30	28	18	16
16	44	40	18	17
32	68	61	19	17
64	112	101	19	17
128	–	175	–	17

Table 4. Number of updates, N_U, to reduce $||\mathbf{H}||_2$ by 10^{-4} for the AF single-grid (SG) and multigrid (MG) schemes; $|\theta_y| \leq 2\pi, \kappa_t = \kappa_d = -3, \alpha = 45\,\text{deg}$

Method	N_x	$\|\|\mathbf{G}\|\|_2$	$\|\|\mathbf{H}\|\|_2$	$\rho(\mathbf{G})$
SG	8	.96	.95	.79
SG	16	.99	.99	.79
SG	32	≈ 1	≈ 1	.80
SG	64	≈ 1	≈ 1	.80
MG	8	1.15	1.02	.79
MG	16	1.76	1.74	.81
MG	32	2.65	2.77	.81
MG	64	4.05	4.01	.82

Table 5. Norms after one update for the AF scheme; $|\theta_y| \leq 2\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

N_x	SG Pred.	SG CFL3D	MG Pred.	MG CFL3D
8	47	37(29)	40	37(26)
16	64	51(45)	44	41(30)
32	91	70(67)	50	43(39)
64	139	106(104)	64	51(49)
128	–	170(165)	–	65(67)

Table 6. Number of updates, N_U, to reduce $||\mathbf{H}||_2$ by 10^{-4} for the AF single grid (SG) and multigrid (MG) schemes; $|\theta_y| \leq 2\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

both with and without multigrid. The CFL number is selected as $1 + \tan\alpha$ because then $\rho(\mathbf{G}) = 0.5$, independent of the mesh. The norms shown in Table 3 after one cycle ($N_c = 1$) indicate a spectral norm for multigrid that is actually slightly *higher* than that of the single grid. The L_2 norms are greater than unity for the multigrid scheme; the single-grid norms are all below unity. The number of updates to reach convergence of $||\mathbf{H}||_2$ below 10^{-4}, with θ_y over all possible discrete frequencies, is shown in Table 4; several convergence calculations were not performed for the highest grid density and are denoted with a $-$ in the Tables. Also shown are the results on a square domain from the baseline CFL3D code[14] with freestream values imposed at the boundaries $x = 0$ and $x = 1$ and with periodicity imposed in the y direction. A random perturbation was imposed in the interior to the density field only, so that the full system of equations emulates the scalar convection equation analyzed here. The Mach number was 0.5 for the CFL3D computations but because the residual equations recognize a contact discontinuity exactly, the results are independent of the Mach number; the only modification required for correspondence was to the time step, because the system of equations bases the time step on the maximum eigenvalue of the full system. For the multigrid computation in either case, the usual W(1,0) cycle is used.

The first-order comparisons show that even though the asymptotic rate is lower with the single-grid scheme, the effective convergence is much better for the multigrid scheme. The single-grid scheme shows a clear dependence on the mesh size and the multigrid rate is nearly grid-independent in both the analysis and the computation. The predicted bounds for the number of cycles required correlate well with the CFL3D results.

Some calculations were made using a residual tolerance of 13 orders of magnitude—an extreme value—to illustrate the slow convergence with iteration of the effective error norm to its asymptotic value given by the spectral radius. With the $\kappa_d = \kappa_t = -3$ scheme, the residual reduction between the last two cycles for the single-grid scheme was $0.55, 0.59, 0.63, 0.68$ for $N_x = N_y = 8, 16, 32, 64$, respectively, as compared to the spectral norm of 0.5.

Next we consider the convergence of the second-order scheme $\kappa_t = 0$ with the first-order implicit AF scheme ($\kappa_d = -3$) for $\alpha = 30\,\text{deg}$, both with and without multigrid. The CFL number is again selected as $1 + \tan\alpha$. The initial norms are shown in Table 5 and indicate a spectral norm for multigrid that is again slightly higher than that of the sin-

N_x	AF	ALJ	ALRB
8	5	3	2
16	5	4	3
32	5	4	3
64	5	4	3

Table 7. Number of updates, N_U, to reduce $||\mathbf{G}^*||_2$ by 1/10 for multigrid (MG) with various relaxation schemes; $|\theta_y| \leq 2\pi$, $\kappa = -3, \alpha = 30\,\text{deg}$

N_x	AF	ALJ	ALRB	Unfactored
8	15	11	9	7
16	19	16	10	8
32	25	21	13	10
64	31	39	21	12

Table 8. Number of updates, N_U, to reduce $||\mathbf{G}||_2$ by 1/28 for defect-correction multigrid (MG) with various relaxation schemes; $|\theta_y| \leq 2\pi$, $\kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

gle grid. The predictions of the analysis with corresponding results from CFL3D are shown in Table 6. Two sets of computations are shown; the results in parentheses are the results with freestream conditions imposed along *all* boundaries. The single-grid analyses and computations show a clear dependence on the mesh size; as we expect, the corresponding multigrid performance is not quite grid-independent, because even the unfactored scheme shows such a dependence. For both of these κ values, the predicted bounds for the number of cycles required correlate well with the numerical results. The number of cycles without periodicity are generally lower than with periodicity imposed, but approaches the same number of cycles as the grid is refined. Although not shown here, similar results were obtained with the $\kappa_t = -1$ formulation; the only differences were that the number of cycles was approximately 20 percent lower with this scheme, as might be expected because the dissipation levels are higher.

First-Order Multigrid Scheme

As a model for the convergence of the linear implicit matrix equation, we consider the multigrid scheme with $\kappa_t = \kappa_d = -3$ and with various relaxation schemes. Table 7 shows results corresponding to a V(1,0) FAS multigrid cycle. Because this particular model is a linear equation with consistent target and driver schemes, the FAS cycle is equivalent to a CS cycle. It is clear that *all* of the schemes converge rapidly and show little variation in the number of iterations to reach convergence as the mesh is refined. Although not shown here, single-grid calculations showed a clear doubling of the number of iterations on each successive mesh refinement.

Second-Order Multigrid Scheme

The convergence for the multigrid scheme with no subiterations is shown in Table 8; the parameters are the same as those for the results in Table 7 except that $\kappa_t = 0$. There is some dependence of the number of cycles to reach convergence, as expected, because even if we eliminate the factorization errors, there is dependence on the mesh density (Table 1).

	N_x	N_s=1	N_s=2	N_s=3	N_s=5
SG	8	12	18	24	40
SG	16	23	24	30	50
SG	32	30	32	42	65
SG	64	57	50	63	95
MG	8	11	16	21	35
MG	16	16	16	24	40
MG	32	21	24	30	50
MG	64	39	36	42	65

Table 9. Number of updates, N_U, to reduce $||\mathbf{G}||_2$ by 1/28 for the ALJ single-grid (SG) and multigrid (MG) schemes; $|\theta_y| \leq 2\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\text{deg}$

The best performance is attained with the ALRB scheme; it degrades little from the unfactored multigrid scheme on coarser meshes. As the mesh is refined, the AF scheme is more competitive because its performance degrades less as the mesh is refined; on the $N_x = 64$ mesh, it is actually more efficient than the ALJ scheme.

Effect of Subiterations, ($N_s > 1$)

We consider the effect of subiterations for the defect-correction scheme corresponding to $\kappa_t = 0$ for $\alpha = 30\,\text{deg}$; Tables 9–10 show results for both the single-grid and the multigrid schemes with ($\nu_1 = 1, \nu_2 = 0$) and with the ALJ scheme. Regarding the error reduction, there is some benefit of a few subiterations, especially as the mesh is refined. Too many subiterations are clearly not efficient when the total number of updates, as used here, is considered. These results are in qualitative agreement with practical calculations for the full systems of equations because $N_s = 3$ provided good performance for the results shown subsequently. In those situations, using 2 subiterations was optimal when stability was maintained but was not robust for large time steps. As shown in Table 10, for the residual reduction no benefit appears at all; the fastest residual reduction occurs with $N_s = 1$, and the disparity with addi-

tional subiterations grows as the mesh is refined. These results are in qualitative agreement with the single-grid local mode analysis of MacCormack and Pulliam[19] for the full system of equations using the defect correction scheme with $\kappa_t = 1/3$; a few subiterations drove the spectral norms below unity.

Efficiency Comparisons

All of the relaxation/smoothers considered here are amenable to vectorization and parallel implementation on computers because they are either Jacobi, red-black, or factored schemes.

The operation counts for the ALJ and ALRB methods are only slightly increased over current block approximate-factorization methods. The overhead is the requirement for additional iterations of the linear system before updating the nonlinear system. However, because the measured computer time for the diagonalized line inversions of the AF scheme for two-dimensional simulations is a factor of four less than the corresponding block inversions, the number of updates of the ALJ and ALRB schemes must show an appreciable improvement over the AF scheme to be viable. In general, this is not attained for the isotropic cases considered here. Substantial differences would be expected in cases of high grid anisotropies, because the AF scheme is known not to be optimal in its present form, as studied extensively by Buelow, et al.[24] The operation counts for the block inversions can be reduced substantially by saving the implicit Jacobians and LU decomposition of the line inversions; the computational results indicate the method is not sensitive to updating these Jacobian entries. In the current implementation, for example, the Jacobians and the LU decomposition of the implicit line solutions are computed initially and reused for each of the updates at a given mesh resolution. For a V(2,1) FAS cycle with three subiterations, the block LU decompositons are done once instead of six times. This saves about a factor of two in computer time and makes the ALJ and ALRB schemes competitive with the diagonalized AF for isotropic cases, albeit at the cost of increased storage. Greater re-use could be made at the cost of additional storage.

Large-Scale Computations

Summarized below are the large-scale computations that have been made for several inviscid and viscous flows, including viscous flow over a flat plate and the separated flow over an airfoil.

	N_x	N_s=1	N_s=3	N_s=5
SG	8	26	54	90
SG	16	49	66	105
SG	32	72	81	130
SG	64	82	108	170
MG	8	34	60	90
MG	16	60	66	100
MG	32	68	72	110
MG	64	70	90	140

Table 10. Number of updates, N_U, to reduce $||\mathbf{H}||_2$ by 10^{-4} for the ALJ single-grid (SG) scheme; $|\theta_y| \leq 2\pi, \kappa_t = 0, \kappa_d = -3, \alpha = 30\,\mathrm{deg}$

Bump in a Channel

Extensive calculations were made for the inviscid flow over a 10-percent thick profile with a $\sin^2$ profile in a channel at a Mach number of 0.5. A prediction methodology based on two-grid local mode analysis for a system of equations, similar to that of Mulder,[9] showed that an asymptotic rate of 0.5 per V(1,0) FAS cycle could be attained with the ALJ scheme with either subiterations or CS multigrid applied to the linear system. Numerical calculations confirmed this; systematic variations of the number of subiterations required indicated that three subiterations without the CS were sufficient to allow large time steps on the order of Courant numbers of 100–300. Calculations made with the CS multigrid applied to the linear system allowed larger time steps, but provided no overall advantage in convergence. Good results were obtained with a V(2,1) FAS cycle with three subiterations of the linear system; the convergence rate corresponded to $0.5^{1/3} = 0.8$ per fine-grid update. Convergence to within 5 percent of truncation error, as measured by the computed drag, occurred within two cycles (or equivalently, 9 fine-grid updates).

Ms. Carolyn Dear of Mississippi State University provided some extensive benchmark evaluations of the baseline AF solver during an intern period at NASA Langley during the summer of 1998. The results showed that the baseline solver in CFL3D provided grid-independent convergence rates using flux-vector splitting as the Riemann solver, but that with flux-difference splitting, the results were grid dependent. The remedy was to apply an entropy fix to the steady residual equations so the minimum eigenvalue of the flux Jacobian matrices did not fall below 0.05 of the maximum value.* Interestingly,

*This is the only modification made to the steady residual operator of the baseline solver CFL3D over the course of this work.

the ALJ formulation did not require this modification to obtain grid-independent convergence. This behavior is attributed to the system of equations and could be studied by extending the methodology here to systems. In comparisons of the two approaches that used the FMG approach with this entropy fix applied, the V(2,1) FAS cycle with 3 subiterations of ALJ as the relaxation scheme was competitive with but did not surpass the baseline solver.

NACA 0012 Airfoil

A series of airfoil calculations indicated results similar to the inviscid simulations above. Again convergence of lift and drag was obtained in a few cycles of the ALJ V(2,1) FAS scheme with 3 subiterations. However, the baseline solver was already adequate to provide efficient multigrid solutions.

For calculations from impulsive freestream initiations, instead of through an FMG process, the use of large time steps presented some difficulties; to overcome these difficulties, the calculation was started at moderate Courant numbers and ramped to large time steps over a few cycles. For the impulsive start, there was a decided improvement over the baseline scheme of the ALRB (or ALJ) subiteration multigrid scheme.

Flat Plate Boundary Layer

The simulation of viscous flow over a flat plate was done for a range of Reynolds numbers with a computational domain extending from $x = -1$ to $x = 2$, with no-slip conditions imposed starting at $x = 0.5$. Constant-pressure boundary conditions were used downstream, along with specified total-pressure, entropy, and velocity-direction boundary conditions at inflow. The aforementioned two-grid local mode analysis indicated that, if the linear system was solved either through subiteration or CS multigrid, the same convergence rate of 0.5 per V(1,0) cycle could be attained. Computations confirmed the analysis; the calculation was insensitive to the grid stretching, and the convergence rate of the residual was better than 0.55 per V(1,0) cycle.

As the grid was stretched, however, it became more difficult to start the solution. Large time steps could be taken, but the increase from small values had to occur slowly. This difficulty was remedied by applying an entropy fix to the implicit side of the equations; the minimum eigenvalue was constrained to be on the order of 0.1 of the maximum eigenvalue. With this modification, large time steps could be taken from impulsive starts. When the CS multigrid was used for the implicit system, the convergence rate of the linear system was better than 0.2 per W(2,1) CS cycle; thus one W(2,1) CS cycle was quite sufficient to solve the linear system to a tolerance of approximately one order of magnitude. The same overall efficiency of the nonlinear residual convergence could be attained if only 3 subiterations, instead of the CS multigrid, were used for the linear system.

In comparisons with the baseline scheme, a calculation was made on a very highly stretched mesh for laminar flow starting from freestream values. To attain convergence of the integrated drag coefficient, the V(2,1) FAS cycle with 3 subiterations of either the ALJ or the ALRB scheme showed a factor of ten reduction in computer time over the baseline scheme The local skin friction values on the plate converged in just a few cycles with this scheme, consistent with the findings of Koren[12] using a CS multigrid defect correction scheme. With an FMG cycle, the improvement was less—approximately a factor of five. Calculations were made using an algebraic turbulence model that showed a similar improvement over the baseline scheme.

Airfoil with Laminar Separation

The convergence efficiency for the flow over a NACA 0012 airfoil at a Mach number of 0.8, $\alpha =$ 10 deg, and a Reynolds number of 500 was investigated for a series of meshes. For this simulation, the region of separation extends over most of the airfoil upper surface, from $x/c = 0.35$ to $x/c = 0.97$. Generally, the lift and drag converged in only a few V(2,1) FAS cycles on meshes varying from 65×25 to 641×129. The grid was a C-type mesh; in order to obtain grid-independent convergence, it was necessary to construct the implicit lines in the wake so that they spanned the wake. The asymptotic convergence was virtually constant on all meshes; reduction of the residual ten orders of magnitude was attained in 40 V(2,1) FAS cycles, corresponding to 120 fine-grid updates using 3 subiterations of the ALRB or the ALJ scheme for each relaxation.

Conclusions

A hierarchical multigrid algorithm for efficient steady solutions to the two-dimensional compressible Navier-Stokes equations has been developed and demonstrated. The general algorithm applies FAS multigrid to a nonlinear target residual equation and CS multigrid to a linearized defect correction implicit equation. The computational work scales as the total number of unknowns in the simulation, N, times the square of the number of equations at each grid point, m, because solutions to block-tridiagonal matrices of block size m are needed. Multigrid analyses that include the effect of boundary conditions

in one direction are used to estimate the convergence rate of the algorithm for a model convection equation. Three alternating-line-implicit algorithms are compared in terms of efficiency. The analyses indicate that full multigrid efficiency is not attained in the general case; the number of cycles to attain convergence is dependent on the mesh for high-frequency cross-stream variations. Of the three algorithms investigated, the baseline AF solver provided the overall best for isotropic grids, considering that its cost per update with the diagonal version is a factor of four cheaper than the full block inversions associated with the ALJ and ALRB schemes. Numerical simulations for a series of flows indicated the ALJ and ALRB were only competitive with the baseline scheme for inviscid flows but were clearly superior for highly stretched viscous mesh simulations. With a V(2,1) FAS cycle with 3 subiterations of either the ALJ or ALRB schemes, convergence of lift and drag to within truncation error occurred in two multigrid cycles.

Acknowledgement

Ms. Carolyn Dear of Mississippi State University provided some extensive benchmark evaluations of the baseline AF solver that proved invaluable in this study during an intern period at NASA Langley in the summer 1998.

References

1. Brandt, A., "Guide to Multigrid Development," *Multigrid Methods, Lecture Notes in Mathematics*, Vol. 960, Springer-Verlag, 1984, pp. 220–317.

2. Brandt, A., "Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD," *ICASE Interim Report No. 32*, NASA/CR-1998-207647, 1998.

3. Roberts, T., Sidilkover, D., and Swanson, R. C., "Textbook Multigrid Efficiency for the Steady Euler Equations," AIAA 97-1949, 1997.

4. Brandt, A. and Yavneh, I., "On Multigrid Solution of High-Reynolds Incompressible Entering Flows," *Journal of Computational Physics*, Vol. 101, 1992, pp. 151–164.

5. Anderson, W. K., Thomas, J. L., and Whitfield, D. L., "Multigrid Acceleration of the Flux Split Euler Equations," AIAA 86-0274, 1986.

6. Jameson, A., and Baker, T. J., "Multigrid Solution of the Euler Equations for Aircraft Configurations," AIAA 85-0122, 1985.

7. Koren, B., "Defect Correction and Multigrid for an Efficient and Accurate Computation of Airfoil Flows," *Journal of Computational Physics*, Vol. 77, 1988, pp. 183–206.

8. Mulder, W. A., "Multigrid Relaxation for the Euler Equations," *Journal of Computational Physics*, Vol. 60, 1985, pp. 235–252.

9. Mulder, W. A., "A Note on the Use of Symmetric Line Gauss-Seidel for the Steady Upwind Differenced Euler Equations," *SIAM J. Sci. Stat. Comp.*, Vol. 11, No. 2, 1990, pp. 389–397.

10. Hemker, P. W., and Spekreijse, S. P., "Multiple Grid and Oshers Scheme for the Efficient Solution of the Steady Euler Equations," *Applied Numerical Mathematics*, Vol. 2,1986, pp. 475–493.

11. Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, 1996, pp. 391–408

12. Koren, B., "Multigrid and Defect Correction for the Steady Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 87, 1990, pp. 25–46.

13. Vatsa, V. N., and Wedan, B. W., "Development of a Flexible and Efficient Multigrid Code for 3-D Navier-Stokes Equations and its Application to a Grid-Refinement Study," *Computers and Fluids*, Vol. 18, No. 4, 1990, pp. 391–403.

14. Krist, S. L., Biedron, R. T., and Rumsey, C. L., "CFL3D User's Manual (Version 5.0)," NASA TM-1998-208444, June 1998.

15. Thomas, J. L., "A Perspective of Computational Fluid Dynamics Research at NASA," *Numerical Methods for Fluid Dynamics V*, Oxford Press, Oxford, 1995, pp. 19–36.

16. Mavriplis, D., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," AIAA 97-1952, 1997.

17. Pierce, N., Giles, M., Jameson, A., and Martinelli, L., "Accelerating Three-Dimensional Navier-Stokes Calculations," AIAA 97-1953, 1997.

18. Allmaras, S., "Algebraic Smoothing Analysis of Multigrid Methods for the 2-D Compressible Navier-Stokes Equations," AIAA 97-1954, 1997.

19. MacCormack, R. W., and Pulliam, T. A., "Assessment of a New Numerical Procedure for Fluid Dynamics," AIAA 98-2821, 1998.

20. Thomas, J. L., Van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA 85-1680, 1985.

21. Pulliam, T. H., and Chausee, D. S., "A Diagonalized Form of an Implicit Approximate Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, 1981, pp. 347–363.

22. Rumsey, C. L., Sanetrik, M. D., Biedron, R.T., Melson, N. D., and Parlette, E. B., "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations," *Computers and Fluids*, Vol. 25, No.2, 1996, pp. 217–236.

23. Brandt, A., "Rigorous Quantitative Analysis of Multigrid I: Constant Coefficients Two-Level Cycle with L_2-Norm," *SIAM J. Numer. Anal.*, Vol. 31, No. 6, 1994, pp. 1695–1730.

24. Buelow, P. E. O., Vankataswaran, S., and Merkle, C. L., "The Effect of Grid Aspect Ratio on Convergence," *AIAA Journal*, Vol. 32, No. 12, 1994, pp. 2401–2408.

25. Desideri, J.-A., and Hemker, P. W., "Convergence Analysis of the Defect-Correction Iteration for Hyperbolic Problems," *SIAM J. Sci. Comput.*, Vol. 16, No. 1, 1995, pp. 88–118.

26. Allmaras, S., Private Communication.

Appendices

I Implementation of ALJ Scheme

The ALJ scheme is implemented with two sweeps through the mesh, as below for the scalar convection equation on mesh h.

$$(\frac{I}{\Delta t} + \delta_x + t\,\hat{\delta}_y^D)_d(\Delta e^*) = -L_t\,(e)^n \qquad (49)$$

$$(\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y)_d(\Delta e) = -L_t\,(e)^n - (\delta_x - \delta_x^D)_d(\Delta e^*) \qquad (50)$$

where the superscript D denotes the diagonal contribution of the operator. Substituting from Eq(49) for $(\delta_x)_d(\Delta e^*)$ into Eq. (50), then

$$(\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y)_d(\Delta e) = (\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y^D)_d(\Delta e^*)$$

Now substituting from Eq. (49) again, the scheme can be written as an approximate factorization scheme, referred to as the DDADI scheme and used by MacCormack and Pulliam[19]

$$(\frac{I}{\Delta t} + \delta_x + t\,\hat{\delta}_y^D)_d(\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y^D)_d^{-1} (\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y)_d(\Delta e^*) \;=\; -L_t\,(e)^n$$

The scheme above is quite similar to the damped ALJ scheme proposed by Mulder.[9] In Mulder's work, as in the relaxation methods used by Thomas, Walters, and Van Leer,[20] the solution and target residuals are updated after each sweep. In the context of the convection scheme studied here, the two sweeps correspond to

$$(\delta_x + 2\,t\,\hat{\delta}_y^D)_d(e^{n+1} - e^n) = -L_t\,(e)^n \qquad (51)$$

$$(2\,\delta_x^D + t\,\hat{\delta}_y)_d(e^{n+2} - e^{n+1}) = -L_t\,(e)^{n+1} \qquad (52)$$

Even though no time step explicitly occurs, the two left sides of Eqs (51-52) are equivalent to the left sides of Eqs (49-50) if, in the latter, we use a CFL of $(t+1)/t$ and $t+1$ on the x-implicit and y-implicit sweeps, respectively. Thus, for non-grid-aligned flows, the time steps are not much greater than those of the AF scheme. Within such a correspondence, if the target scheme is the same as the implicit scheme, as for example with subiterations or with a first-order-accurate scheme, these two approaches would be identical. In the general case, the two schemes would behave differently because of the above differences in the way the first sweep influences the second sweep.

II Implementation of ALRB Scheme

The ALRB scheme is implemented very similarly to the above scheme, except that the implicit lines are updated in a red-black fashion, as below, assuming that $\kappa_d = -3$.

$$(\frac{I}{\Delta t} + \delta_x + t\,\hat{\delta}_y^D)_d(\Delta e^*) = -L_t\,(e)^n \qquad : B_x \qquad (53)$$

$$(\frac{I}{\Delta t} + \delta_x + t\,\hat{\delta}_y^D)_d(\Delta e^*) = -L_t\,(e)^n - (t\,\hat{\delta}_y - t\,\hat{\delta}_y^D)(\Delta e^*) \qquad : R_x \qquad (54)$$

$$(\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y)_d(\Delta e^{**}) = -L_t\,(e)^n - (\delta_x - \delta_x^D)_d(\Delta e^*) \qquad : B_y \tag{55}$$

$$(\frac{I}{\Delta t} + \delta_x^D + t\,\hat{\delta}_y)_d(\Delta e^{**}) = -L_t\,(e)^n - (\delta_x - \delta_x^D)_d(\Delta e^{**}) \qquad : R_y \tag{56}$$

where (R_x, B_x) and (R_y, B_y) refer to a sequencing of the (x, y) lines in red-black fashion. Now Eqs (53) and (54) can be written as

$$(N_x)_d(\Delta e^*) = -L_t\,(e)^n \tag{57}$$

where N_x denotes the lower triangular part of the full driver operator $I/\Delta t + L_d$ after a resequencing of the $x =$ constant lines in a red-black fashion. Likewise the second set of equations, by adding and subtracting the term $L_d(\Delta e^*)$ to the right-hand sides, can be written as

$$(N_y)_d(\Delta e^{**} - \Delta e^*) = -L_t\,(e)^n - L_d(\Delta e^*) \tag{58}$$

where N_y denotes the lower triangular part of the full driver operator $I/\Delta t + L_d$ after a resequencing of the $y =$ constant lines in a red-black fashion. A composite operator can then be written as

$$\begin{aligned} \Delta e^{**} &= -(N_y^{-1} + N_x^{-1} - N_y^{-1} L_d N_x^{-1}) L_t\,(e)^n \\ &\equiv -(\overline{L_d})^{-1}\,L_t\,(e)^n \end{aligned} \tag{59}$$

III Amplification Matrix for ALRB

The amplification matrix for the step corresponding to Eq. (53) can be written as

$$\mathbf{G}^1 = \frac{1}{2}\,\mathcal{C}(\mathbf{G}^{B1}, \mathbf{G}^{R1}) \tag{60}$$

where

$$\mathcal{C}(\mathbf{G}, \mathbf{H}) \equiv \mathbf{G} + \mathbf{H} + \begin{bmatrix} \mathbf{G}_{21} - \mathbf{H}_{21} & \mathbf{G}_{22} - \mathbf{H}_{22} \\ \mathbf{G}_{11} - \mathbf{H}_{11} & \mathbf{G}_{12} - \mathbf{H}_{12} \end{bmatrix} \tag{61}$$

and where $\mathbf{G}^{R1} = \mathbf{I}$,

$$\mathbf{G}^{B1} = \mathbf{I} - \mathbf{M}^{-1}\,\mathbf{L}_t \tag{62}$$

and $\mathbf{M}$ is a diagonal matrix corresponding to the x-implicit/y-Jacobi approximation, i.e., $M = (I/\Delta t + \delta_x + t\,\hat{\delta}_y^D)_d$ along the diagonal entries. The amplification matrix for the step corresponding to Eq. (54) can be written as

$$\mathbf{G}^2 = \frac{1}{2}\,\mathcal{C}(\mathbf{G}^{B2}, \mathbf{G}^{R2}) \tag{63}$$

where $\mathbf{G}^{B2} = \mathbf{G}^1$,

$$\mathbf{G}^{R2} = (\mathbf{I} - \mathbf{M}^{-1}\mathbf{L}_d)\mathbf{G}^1 + \mathbf{M}^{-1}(\mathbf{L}_d - \mathbf{L}_t) \tag{64}$$

The amplification matrix after the completion of the y-implicit/x-RB sweep corresponding to Eqs. (55) and (56) can be written as

$$\overline{\mathbf{G}} = (\mathbf{I} - \mathbf{N}_y^{-1}\,\mathbf{L}_d)\mathbf{G}^2 + \mathbf{N}_y^{-1}(\mathbf{L}_d - \mathbf{L}_t) \tag{65}$$

where $\mathbf{N}_y$ is the matrix corresponding to a y-implicit/x-RB approximation to $\mathbf{L}_d$. There is some dependence of the iteration on the order of the points taken; here, the first sweep is for the odd points, which includes the point closest to the inflow boundary.

IV Restrictions and Prolongations

The restriction and prolongation matrices R and P associated with the x-direction coarsening are of dimensions $N_x/2 \times N_x$ and $N_x \times N_x/2$, respectively, as below.

$$R = \begin{bmatrix} r_1 & r_1 & r_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_1 & r_1 & r_2 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & r_2 & r_1 & r_1 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2 & r_1 & r_1 \end{bmatrix}$$

$$P = \begin{bmatrix} p_1 & p_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p_1 & p_2 & p_3 & 0 & 0 & 0 & 0 & 0 \\ p_2 & p_1 & p_0 & 0 & 0 & 0 & 0 & 0 \\ p_0 & p_1 & p_2 & p_3 & 0 & 0 & 0 & 0 \\ p_3 & p_2 & p_1 & p_0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & p_0 & p_1 & p_2 \\ 0 & 0 & 0 & 0 & 0 & p_3 & p_2 & p_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_0 & p_1 \end{bmatrix}$$

where for the usual prolongations and restrictions

$$\{p_0, p_1, p_2, p_3\} = \frac{1}{4}\{0, 3, 1, 0\}$$

$$\{r_1, r_2\} = \frac{1}{2}\{1, 0\}$$

and for the higher-order prolongations and restrictions, denoted with asterisks,

$$\{p_0, p_1, p_2, p_3\} = \frac{1}{384}\{-21, 315, 105, -5\}$$

$$\{r_1, r_2\} = \frac{1}{6}\{4, -1\}$$

V A Curious Case of Norms

As an illustration of the difference between the L_2 and spectral norms, the convergence for grid-aligned flow, $t = 0$, is considered. With the $\kappa = -1$ differencing scheme, Desideri and Hemker[25] have shown that the predicted asymptotic error decay rate is very fast, $\rho(\mathbf{G}) \leq 0.5$, but that the asymptotic rate is achieved only after $2N_x$ cycles. This odd behavior is associated with a deficient set of eigenvectors of the iteration matrix; the initial slow decay of the residual over the first $2N_x$ iterations is termed a pseudo-convection phase by Desideri and Hemker.[25] The L_2 norms of $\mathbf{G}$ and $\mathbf{H}$ remain close to unity for $2N_x$ cycles and then fast convergence is attained. Because the left and right side matrices are lower triangular, it is straightforward to show that the asymptotic spectral radius is determined by the ratio of the diagonal terms between the driver and target schemes, i.e.,

$$\rho(\mathbf{G}) = |1 - \frac{(\delta_x)_t^D}{(\delta_x)_d^D}|$$

By scaling the left side implicit matrix by $3/2$, equivalent to an underrelaxation[26] of the right side residual by $2/3$, the spectral norm becomes zero and the L_2 norms of both $\mathbf{G}$ and $\mathbf{H}$ are $1/3$. Now the convergence is quite fast—so fast, in fact, that the asymptotic rate of zero, which is indeed achieved after N_x cycles, is usually not seen before machine zero is reached.